JOURNAL OF COMPUTER AND SYSTEM SCIENCES 36, 379–383 (1988)

Expanders, Randomness, or Time versus Space

MICHAEL SIPSER*

Mathematics Department, MIT, Cambridge, Massachusetts 02139

Received December 11, 1986; revised June 25, 1987

Let *EH* be the hypothesis that a certain type of expander graph has an explicit construction. Let io-SPACE($T(n)$) be the class of problems solvable by algorithms that for infinitely many inputs use at most space $T(n)$. Then the following holds: There exists $\varepsilon > 0$ such that for any polynomial time bound $T(n) = n^k$,

$$EH \rightarrow (\mathrm{P} = \mathrm{R} \text{ or } \mathrm{TIME}(T(n)) \subseteq \text{io-SPACE}(T^{1-\varepsilon}(n))).$$

1. INTRODUCTION

Our main result follows from two theorems described informally in the next two subsections and more carefully in Section 3.

1.1. *Randomness as a Resource*

One measure of the efficiency of a randomized algorithm is the number of random bits used. For example, the obvious method of amplifying the probability of success of an R-machine is to run it many times with independently made random choices. This decreases the probability of missing an accepting computation when there is one, but it appears wasteful in that it uses each random bit exactly once and then discards it.

We describe a connection between explicit construcion of certain types of expander graphs and the ability to achieve the above type of amplification using fewer random bits.

1.2. *Time versus Space and Pseudorandomness*

The beautiful work of Blum and Micali, Yao, and others [BM, Y] has shown how good pseudorandom number generators admit subexponential deterministic algorithms for problems in R. We show that either all time-bounded computations have somewhat more efficient space-bounded simulations or one can generate sequences that are akin to pseudorandom sequences in that they can be used for

* Research supported in part by NSF Grant DCR-8602062 and Air Force Grant AFOSR-86-0078. Some of this work was done while the author was on the faculty of the University of California/Berkeley and visiting the Mathematical Science Research Center at Berkeley.

0022-0000/88 $3.00

improved deterministic simulations of certain randomized algorithms. This is related to an earlier paper of Hopcroft, Paul, and Valiant [HPV]. They prove that $\mathrm{TIME}(t(n)) \subseteq \mathrm{SPACE}(t(n)/\log(t(n))$ using the pebbing game. Our result gives a somewhat better bound than does theirs but only holds in the presence of unproven hypotheses regarding expander graphs and the P vs. R question.

2. Expander Graphs

Expanders are a type of graph with numerous applications; see [K] for extensive discussion. It is often relatively straightforward to prove the existence of such graphs by non-constuctive methods [Pin, P]. In some cases, explicit constructions are known [M, GG, LPS]. The types of expanders needed in this paper are only known to exist non-constructively.

Definition. An (l, r, d, k)-expander is a bipartite graph with l left nodes each of degree d, and r right nodes, in which every subset of k left nodes *covers* (i.e., is connected to) more than $r/2$ right nodes.

Theorem. *For any m, there exist $(m^{\log m}, m, 2\log^2 m, m)$-expanders.*

Proof. Probabilistic construction. Select the $2\log^2 m$ edges randomly at each left node by choosing a random set of size $2\log^2 m$ from the m right nodes. Then:

$$\begin{aligned}
&\Pr[\exists m \text{ left nodes attached to } \leqslant m/2 \text{ right nodes}] \\
&\leqslant \binom{m^{\log m}}{m} 2^{-2m\log^2 m} \binom{m}{m/2} \\
&\leqslant m^{m\log m}\, 2^{-2m\log^2 m}\, 2^m \\
&\leqslant 2^{m\log^2 m}\, 2^{-2m\log^2 m}\, 2^m \\
&\leqslant 2^{m - m\log^2 m} \ll 1
\end{aligned}$$

Remark. These expander graphs also have the property that every set of $r/2$ right nodes covers more than $l-k$ left nodes. If not, then the uncovered left nodes would contradict the expansion property.

2.1. *Deterministic Expander Graphs*

Let $m = 2^q$. Let the 2^q right nodes of the aforementioned expander graphs be labeled with the strings from Σ^q ($\Sigma = \{0, 1\}$) and the 2^{q^2} left nodes labeled with the strings in Σ^{q^2}. Say that a family of expander graphs G_q, $q = 1, 2, \ldots$, has an explicit construction if there is a polynomial time computable function which, given the label of a left node in G_q, returns the collection of labels of adjacent right nodes. It is not known if such a family exists. Let the *expander construction hypothesis* (EH) state that there is such a family.

3. Main Results

3.1. *Deterministic Expanders and Strong-R*

Let M be an R-machine accepting language W. Say that on inputs of length n, M uses $q(n)$ random bits. For $w \in \Sigma^n$, $\rho \in \Sigma^{q(n)}$ write $M(w, \rho) =$ **accept** or **reject** to mean M on input w with random sequence ρ accepts or rejects. The error probability of M on input $w \in W$ is $|B|/2^{q(n)}$ where $B = \{\rho: M(w, \rho) = \textbf{reject}\}$. The error probability of M is the maximum error probability of M on any w.

Conventional R-machines are defined to have error probability at most $\frac{1}{2}$. Using $p(n)$ repeated independent simulations this may be brought down to $2^{-p(n)}$.

THEOREM. *If EH is true then, for any* R*-machine M using $q(n)$ random bits with error probability $\frac{1}{2}$, there is another* R*-machine N using $q^2(n)$ random bits with error probability* $2^{-(q^2(n)-q(n))}$.

Proof. $N(w, \rho)$ treats $\rho \in \Sigma^{q^2(n)}$ as a left node in an expander and simulates M with the $2q^2(n)$ right nodes to which ρ is connected. The improvement in error probability follows directly from the remark preceeding Section 2.1. This is because one may take the set of accepting computations of M as selecting a set of $2^{q(n)}/2$ right nodes. By the remark these cover at least $2^{q^2(n)-q(n)}$ left nodes. N will fail to find an accepting computation if it selects one of the at most $2^{q(n)}$ uncovered left nodes. The probability of this occurring is at most $2^{q(n)}/2^{q^2(n)} = 2^{(q(n)-q^2(n))}$.

DEFINITION. Let $A \in$ *strong-R* if there is an R-machine accepting A using random sequences of length $q(n) = n^j$ for some j and with error probability $2^{-(q(n)-q^\alpha(n))}$ for some $\alpha < 1$.

COROLLARY. EH $\rightarrow$ (R = strong-R).

The parameters in EH may be weakened somewhat while maintaining the above corollary. For example, it is enough to have explicit constructions for $(m^{\log m}, m, \log^j m, m^{(\log m)/2})$-expanders for any fixed j.

3.2. *P = Strong-R or Time versus Space*

THEOREM. *One of the following holds*:

(a) $P =$ *strong-R or*

(b) $\exists \varepsilon > 0$, *for any polynomial time bound* $T(N) \geqslant N$, $\text{TIME}(T(N)) \subseteq \text{io-SPACE}(T^{1-\varepsilon}(N))$.

Proof. If P $\neq$ strong-R then let $A \notin$ P be accepted by strong-R machine S. On inputs of length n, S uses $q(n) = n^j$ random bits, has error probability $2^{-(q(n)-q^\alpha(n))}$ for some $\alpha < 1$, and runs in time $t(n)$. Assume that $t(n) = n^l$. Let $\varepsilon = \min(\frac{1}{2}, (1-\alpha)j/4l)$.

Let $T(N) = N^k$ and let $B \in \text{TIME}(T(N))$ be accepted by M. We construct M' accepting B operating infinitely often in space $T^{1-\varepsilon}(N)$. In particular, we show that M' operates in space $T^{1-\varepsilon}(N)$ on a subset of 1^*.

Machine M': Input 1^N. Let $\beta = j/4l$.
For all circuits C_i on $T^\beta(N)$ inputs, describable in space $T^{1-\varepsilon}(n)$:

1. Let $D = \{x\colon C_i(x) = 1\}$. If $\log_2 |D| > T^{\alpha\beta}(N)$, then restart the simulation with circuit C_{i+1}.
2. Prepare the actual simulation of M as follows. Break all tapes into blocks of size $T^\beta(N)$. Represent each block b by the index of b, $i_D(b)$ in the lexicographic ordering of D. If $b \notin D$ for any such b then restart with circuit C_{i+1}.
3. Perform the actual simulation of M as follows. Each "active" block, i.e., one containing a head, is represented explicitly in the conventional fashion. All other blocks are represented in the encoded fashion, as their index within D. Whenever a head crosses a block boundary, the old block, b_{old}, is "closed" by determining its index within D, $i_D(b_{\text{old}})$ and only storing the index. Note that the length of $i_D(b_{\text{old}})$ is at most $T^{\alpha\beta}(N)$. The new block is "opened" by decoding its index through a lookup within D. If at any point the closing procedure fails because $b_{\text{old}} \notin D$ then restart the simulation with C_{i+1}.

If the above uses up all circuits C_i without ever completing the simulation then return "failure."

If the above simulation succeeds, then it runs in space $T(N) \cdot (T^{\alpha\beta}(N)/T^\beta(N)) = T^{1-(1-\alpha)\beta}(N) \leqslant T^{1-\varepsilon}(N)$. If the simulation fails, then the blocks produced by running M on 1^N provide good sequences for simulating probabilistic algorithms. In fact, they would be good enough to simulate the strong-R machine S in deterministic polynomial time, a contradiction. The deterministic simulation follows.

Deterministic simulation of S: On input x of length n, let $N = T^{-1}(n^{4l})$. (We assume here for simplicity that T^{-1} is well defined and integral. If not then small adjustments preserve the argument. That these adjustments are small follows because the running time $T(N) = N^k$ does not increase very sharply at any point.) Run M on input 1^N obtaining all blocks of length $T^\beta(N) = q(n)$. Simulate S using each of these in turn in place of the random input. If an accepting computation is found then accept, otherwise reject.

This is guaranteed to find an accepting path if there is one. Let $C_{S,x}$ be the circuit simulating S on x where the input to $C_{S,x}$ is the $q(n)$ bit random input to S. Its negation is $\neg C_{S,x}$. Note that $\neg C_{S,x}$ may be represented in space $t^2(n) = n^{2l} = T^{1/2}(N) \leqslant T^{1-\varepsilon}(N)$. Additionally, $\neg C_{S,x}$ takes inputs of length $q(n)$ and accepts at most $2^{q^\alpha(n)} = 2^{T^{\alpha\beta}(N)}$ of its inputs. Therefore $\neg C_{S,x}$ would have been one of the circuits occurring in the above simulation of M. Because this simulation failed $\neg C_{S,x}$ does not accept all blocks of M and hence some block causes $C_{S,x}$ to accept. Therefore, if S accepts x then it accepts x using some block of M. Thus the deterministic simulation works as claimed.

ACKNOWLEDGMENTS

Dick Karp first suggested the notion of treating randomness as a resourse several years ago. I am also grateful to Martin Tompa, Johan Hastad, and the referees who commented on earlier versions of this paper. I owe a special debt to Philip Klein who proposed the possibility of improving the Hopcroft, Paul, and Valiant [HPV] construction to me. This paper exists only because of that discussion.

REFERENCES

[P] N. PIPPENGER, Superconcentrators, *SIAM J. Comput.* **6** (197), 298–304.

[K] M. KLAWE, Non-existence of one dimensional expanding graphs, *in* "Proceedings, Conference on Foundations of Computer Science 22, 1981," pp. 109–114.

[Pin] M. PINSKER, On the complexity of a concentrator, *in* "Proceedings, 7th International Teletraffic Conference, Stockholm, 1973."

[GG] O. GABBER AND Z. GALIL, Explicit constructions of linear size superconcentrators, *in* "Proceedings, Conference on Foundations of Computer Science 20, 1979," pp. 364–370.

[M] G. MARGULIS, Explicit constructions of concentrators, *Problems Inform. Transmission* (1975).

[HPV] J. HOPCROFT, W. PAUL, AND L. VALIANT, On time versus space, *J. Assoc. Comput. Mach.* **24** (1977), 332–337.

[BM] M. BLUM AND S. MICALI, How to generate cryptographically strong sequences of pseudo-random bits, *SIAM J. Comput.* **13** (1984), 850–863.

[Y] A. YAO, Theory and application of trapdoor functions, *in* "Proceedings, Conference on Foundations of Computer Science 23, 1982, pp. 80–91.

[LPS] A. LUBOTSKY, R. PHILLIPS, AND P. SARNAK, Explicit expanders and the Ramanujan conjectures, *in* "Proceedings, 18th Annual Conference on the Theory of Computing, 1986," pp. 240–246.